Coinductive resumption monads

Levy, Paul Blain; Goncharov, Sergey

DOI:
10.4230/LIPIcs.CALCO.2019.13

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Levy, PB & Goncharov, S 2019, Coinductive resumption monads: guarded iterative and guarded Elgot. in M Roggenbach & A Sokolova (eds), *8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019)*. vol. 139, Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl, pp. 13:1--13:17, 8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019), London, United Kingdom, 3/06/19. https://doi.org/10.4230/LIPIcs.CALCO.2019.13

Coinductive Resumption Monads: Guarded Iterative and Guarded Elgot

Paul Blain Levy
University of Birmingham, UK
P.B.Levy@cs.bham.ac.uk

Sergey Goncharov
FAU Erlangen-Nürnberg, Germany
Sergey.Goncharov@fau.de

Abstract

We introduce a new notion of "guarded Elgot monad", that is a monad equipped with a form of iteration. It requires every guarded morphism to have a specified fixpoint, and classical equational laws of iteration to be satisfied. This notion includes Elgot monads, but also further examples of partial non-unique iteration, emerging in the semantics of processes under infinite trace equivalence.

We recall the construction of the "coinductive resumption monad" from a monad and endofunctor, that is used for modelling programs up to bisimilarity. We characterize this construction via a universal property: if the given monad is guarded Elgot, then the coinductive resumption monad is the guarded Elgot monad that freely extends it by the given endofunctor.

2012 ACM Subject Classification Theory of computation → Categorical semantics; Theory of computation → Axiomatic semantics

Keywords and phrases Guarded iteration, guarded monads, coalgebraic resumptions

Digital Object Identifier 10.4230/LIPIcs.CALCO.2019.13

Funding *Sergey Goncharov*: Support by Deutsche Forschungsgemeinschaft (DFG) under project GO 2161/1-2 is gratefully acknowledged.

1 Introduction

The study of monads for effects has developed in numerous directions since it was initiated in [18]. We make two contributions to this research area. Firstly we give a new notion of "guarded Elgot monad" – a monad equipped with a form of iteration – that includes a variety of examples. Secondly, we give a universal property for one of these examples, the so-called "coinductive resumption monad". We shall explain these contributions separately.

1.1 Monads and Iteration

Monads. Let us recall the basic ideas of monads for effects, where the base category is **Set**. A monad $\mathbf{T}$ on **Set**, presented in "Kleisli triple" form, consists of three things.

- For each set X, a set TX, of which an element represents a "computation" that may perform various computational effects and may return an element of X.
- For each set X, a map $\eta_X \colon X \to TX$. For $x \in X$, the image $\eta_X(x)$ represents a "pure computation" that just returns x.
- For any map $f \colon X \to TY$, we have a map $f^\star \colon TX \to TY$. For $p \in TX$, the image $f^\star(p)$ represents a "sequenced computation" that first executes p and then, if this returns $x \in X$, proceeds to execute $f(x) \in TY$.

These must satisfy three equations, as described in [18]. A map $X \to TY$ is called a *Kleisli map*, and these form the *Kleisli category*, denoted $\mathrm{Kl}(\mathbf{T})$. It inherits coproducts from **Set**.

Monads for printing. We give (in outline) some example monads for computations that print characters. Let A be an alphabet, i.e. a set of characters. We write $A^\star$ (resp. A^ω, $A^{\leqslant\omega}$) for the set of finite sequences (resp. infinite sequences, finite and infinite sequences). Here are our examples.

- The monad $X \mapsto A^\star \times X$ represents computations that print several characters and then return a value.
- The monad $X \mapsto A^\star \times X + A^{\leqslant\omega}$ represents such computations, but also computations that continue forever and never return. The latter includes computations that print finitely many characters and then diverge (i.e. hang), and also computations that print infinitely many characters.
- The monad $X \mapsto A^\star \times X + A^\omega$ represents computations that may return or continue forever, but in the latter case are required to be "productive", i.e. keep printing.

For our next series of examples, write $\mathcal{P}^+X$ for the set of nonempty subsets of X. The following are monads for *nondeterministic* printing computations.

$$\begin{aligned} X &\mapsto \mathcal{P}^+(A^\star \times X) \\ X &\mapsto \mathcal{P}^+(A^\star \times X + A^{\leqslant\omega}) \\ X &\mapsto \mathcal{P}^+(A^\star \times X + A^\omega) \end{aligned}$$

A nondeterministic printing computation has *terminating traces* in $A^\star \times X$, *divergences* in $A^\star$ and *infinite traces* in A^ω. (A similar arrangement has been used in CSP semantics [21].) The above monads identify computations that are *infinite trace equivalent*, i.e. that have the same terminating traces, divergences and infinite traces.

Iterative computations. Given a Kleisli map $f\colon X \to Y + X$, we would like to form a Kleisli map $f^\dagger\colon X \to Y$ where, for $x \in X$, the image $f^\dagger(x)$ represents the following "iterative computation". First it executes $f(x) \in T(Y + X)$. That may return $\mathsf{inl}\, y$, in which case the iterative computation returns y, or it may return $\mathsf{inr}\, x'$, in which case the computation represented by $f(x') \in T(Y + X)$ is executed, and so forth. Can we form $f^\dagger$ for our example monads?

- For the monad $X \mapsto A^\star \times X + A^{\leqslant\omega}$, we can form $f^\dagger$, since the monad is able to represent infinite computations.
- For the monad $X \mapsto \mathcal{P}^+(A^\star \times X + A^{\leqslant\omega})$, $f^\dagger$ is formed analogously.
- For the monad $X \mapsto A^\star \times X + A^\omega$, we can form $f^\dagger$, provided f is *guarded*. That means that for all $x \in X$, the image $f(x) \in A^\star \times (X + Y) + A^\omega$ is not of the form $\mathsf{inl}(\varepsilon, \mathsf{inl}\, x')$ for some $x' \in X$. This condition ensures that $f^\dagger(x)$ represents a productive computation, because an iterative call is possible only after at least one character has been printed.
- For the monad $\mathcal{P}^+(A^\star \times X + A^\omega)$, $f^\dagger$ is formed analogously. Here the guardedness requirement is that, for all $x \in X$, the image $f(x) \in \mathcal{P}^+(A^\star \times (X + Y) + A^\omega)$ does not contain $\mathsf{inl}(\varepsilon, \mathsf{inr}\, x')$ for any $x' \in X$.

These four examples motivate the first contribution of the paper, viz. the notion of a *guarded Elgot monad*. This consists of a monad on a co-Cartesian category $\mathcal{C}$ (i.e. category with finite coproducts), equipped with two additional structures. Firstly a *guardedness predicate*, that tells us when a Kleisli map $f\colon X \to Y + Z$ is guarded in the right summand. When this condition holds, we write $f\colon X \to Y \rangle Z$. Secondly, a *guarded Conway operator* that associates to each map $f\colon X \to Y \rangle X$ a Kleisli map $f^\dagger\colon X \to Y$. Each of these structures

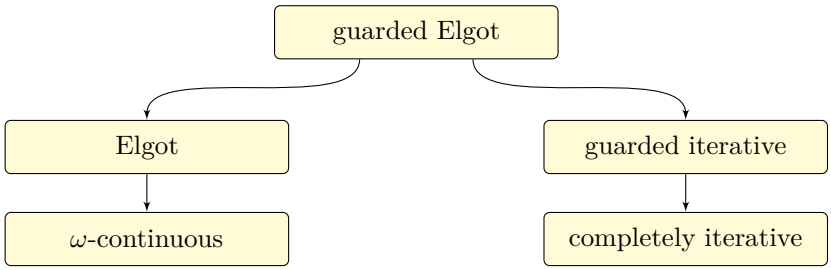
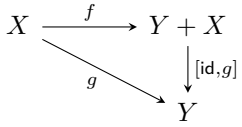


Figure 1 Connections between classes of monads with iteration.

must satisfy some conditions that we shall stipulate. In particular, for $f\colon X \to Y \rangle X$, we require $f^\dagger$ to be a *fixpoint* of f, i.e. a Kleisli map $g\colon X \to Y$ such that



commutes in the Kleisli category.

Although the above four examples are all guarded Elgot monads, they are significantly different.

- The guarded Elgot monads $X \mapsto A^\star \times X + A^{\leqslant\omega}$ and $X \mapsto \mathcal{P}^+(A^\star \times X + A^{\leqslant\omega})$ are special because *every* Kleisli map $f\colon X \to Y + Z$ is deemed to be guarded in the right summand. So, for every Kleisli map $f\colon X \to Y + X$, we can form $f^\dagger$. We call these simply *Elgot monads*. (They are called "complete Elgot monads" in [7].)
- The guarded Elgot monad $X \mapsto A^\star \times X + A^\omega$ is special because, for each map $f\colon X \to Y \rangle X$, the map $f^\dagger$ is the *unique* fixpoint of f. We call this a *guarded iterative monad* [10].
- For $A \neq$, the guarded Elgot monad $X \mapsto \mathcal{P}^+(A^\star \times X + A^\omega)$ is neither Elgot nor guarded iterative. (This is proved in Example 20(5) below). So it illustrates the need for the new, more general notion of guarded Elgot monad.

As noted in [10], *every* monad can be regarded as guarded iterative, by saying that a Kleisli map $f\colon X \to Y + Z$ is "vacuously" guarded in the right summand when it factorizes via $\mathsf{inr}\colon Z \to Y + Z$.

1.2 Resumption Monads

Let us write $\mu\gamma.F\gamma$ for an initial algebra of F, and $\nu\gamma.F\gamma$ for a final coalgebra. We note the following.

- The set $A^\star \times X$ can be written $\mu\gamma.(X + H\gamma)$, where H is the endofunctor $Y \mapsto A \times Y$.
- The set $A^\star \times X + A^\omega$ can be written $\nu\gamma.(X + H\gamma)$.
- The set $A^\star \times X + A^{\leqslant\omega}$ can be written $\nu\gamma.\,\mathsf{Maybe}(X + H\gamma)$, where $\mathsf{Maybe}\ Y \stackrel{\text{def}}{=} Y + 1$.

More generally, given a monad T and endofunctor H on a co-Cartesian category $\mathcal{C}$, we form two monads:

- the *inductive resumption monad* $\mathbf{T}^\mu_H$ sending $X \mapsto \mu\gamma.\,T(X + H\gamma)$, provided these initial algebras exist [6]
- the *coinductive resumption monad* $\mathbf{T}^\nu_H$ sending $X \mapsto \nu\gamma.\,T(X + H\gamma)$, provided these final coalgebras exist [20].

For example, with $\mathcal{C} = \mathbf{Set}$, let T be the countable nonempty powerset monad and let $H\colon Y \mapsto A \times Y$. Then these monads represent countably nondeterministic printing computations modulo bisimilarity. Here, the difference between T^{μ}_{H} and T^{ν}_{H} is that the former represents only computations that eventually return a value, whereas the latter represents also computations that continue forever (but are productive).

For another class of examples, let $(B(a))_{a\in A}$ be a "signature", i.e. family of sets, and let H be the endofunctor $Y \mapsto \sum_{a\in A} Y^{B(a)}$. Again let T be the countable nonempty powerset monad. In this case the monads T^{μ}_{H} and T^{ν}_{H} represent countably nondeterministic computations that perform I/O. Such a computation can print an element $a \in A$ and then pause; if the user then enters an element of $B(a)$, the computation resumes. This is the reason for the name "resumption monad". The printing example is the special case where $B(a)$ is singleton for all $a \in A$.

As the above examples illustrate, these monads provide a natural way of combining an endofunctor (representing I/O) with a monad (representing other effects, e.g. nondeterminism). So one may ask of each monad: can it be characterized via a universal property?

This has been done for T^{μ}_{H} in [15]. We recall this result, but present it a little differently, using the notion of *free extension* (defined in full generality in Definition 2 below).

1.3 Free Extensions

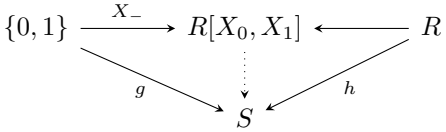
To explain the notion of free extension, we give a well-known example: the polynomial ring $R[X_0, X_1]$. This is the free extension of the ring R by the set $\{0,1\}$. That means that we have a function and ring homomorphism

$$\{0,1\} \xrightarrow{X_-} R[X_0, X_1] \xleftarrow{\quad} R$$

(here X_- reads as a map sending $i \in \{0,1\}$ to X_i) that are universal: for any function and ring homomorphism

$$\{0,1\} \xrightarrow{g} S \xleftarrow{h} R$$

there is a unique mediating homomorphism



We can now describe the result of [14] as follows: the monad T^{μ}_{H} is a free extension of T by H. This means that we have a natural transformation and monad morphism

$$H \xrightarrow{\beta} \mathsf{T}^{\mu}_{H} \xleftarrow{\rho} \mathsf{T}$$

that are universal.

The second contribution of this paper is the following analogous result. If T is a guarded Elgot monad, then T^{ν}_{H} is also guarded Elgot, and moreover it is the free extension, among guarded Elgot monads, of T by H. This means that – in a suitable sense we shall define – we have a guarded natural transformation and guarded Elgot monad morphism

$$H \xrightarrow{\beta} \mathsf{T}^{\nu}_{H} \xleftarrow{\rho} \mathsf{T}$$

that are universal. This in turn gives a universal property for the two special cases simply by varying the notion of guardedness in which our result is parametric.

- If $\mathbf{T}$ is Elgot, then $\mathbf{T}^{\nu}_{H}$ is Elgot, and therefore it is the free extension, among Elgot monads, of $\mathbf{T}$ by H. This result appeared (with a considerably more complex proof) in [9].
- If $\mathbf{T}$ is guarded iterative (as noted above, *any* monad can be so regarded), then $\mathbf{T}^{\nu}_{H}$ is guarded iterative [10], and therefore it is the free extension, among guarded iterative monads, of $\mathbf{T}$ by H. A similar result – using "two-sided ideals" rather than guardedness predicates – was given in [19, Corollary 4.6], generalizing [16].

In general, a free extension of an initial object is a free object. (This is Proposition 4 below.) For example, the ring $\mathbb{Z}$ of integers is initial among all rings, so $\mathbb{Z}[X_0, X_1]$ is a free ring on the set $\{0, 1\}$. This gives some more special cases.

- The identity monad is initial among all monads. So Id^{μ}_{H} is a free monad on H.
- The identity monad is initial among all guarded iterative monads, and among all guarded Elgot monads. So Id^{ν}_{H} is a free guarded iterative monad, and a free guarded Elgot monad, on H. With $H = \mathrm{Id}$ this yields Capretta's *delay monad* $\nu\gamma.\, - + \gamma$ used for modeling partiality in intensional type theory [5].
- On **Set**, the Maybe monad is initial among all Elgot monads. This is true, more generally, on any *hyperextensive category* [1]. So Maybe^{ν}_{H} is a free completely Elgot monad on H. This was previously shown in [9].

It is also worth noting that free extensions can also be described as coproducts with free objects. (This is Proposition 5 below). For example, the free extension of a ring R by the set $\{0, 1\}$ can be described as the coproduct of R and the free ring on $\{0, 1\}$. This formulation is used in [14, 13, 19, 9] and indeed we provide a coproduct characterization in this style in Corollary 29 below. We take the view, however, the characterization in terms of free extensions is more primitive, since it does not require the free object to exist.

2 Preliminaries

In this paper we work in co-Cartesian categories, which are categories with finite coproducts. We fix selected coproduct co-spans $X \xrightarrow{\mathsf{inl}} X + Y \xleftarrow{\mathsf{inr}} Y$ and initial objects 0 with $[\,] \colon 0 \to X$ denoting the initial morphisms. We do not generally assume *extensiveness*, in particular, the injections inl and inr need not be monic.

In a category $\mathcal{C}$, we denote by $|\mathcal{C}|$ the associated class of objects and by $\mathcal{C}(X, Y)$ the set of morphisms from X to Y. We occasionally omit indexes at natural transformation components to improve readability. For a functor $F \colon \mathcal{C} \to \mathcal{C}$, we denote by $(\nu F, \mathsf{out} \colon \nu F \to F\nu F)$ the *final F-coalgebra*. Whenever possible, we use bold letters, e.g. $\mathbf{T}$, for monads, to emphasize the distinction with the underlying functor T. A monad $\mathbf{T}$ over $\mathbf{C}$ induces a *Kleisli category* $\mathrm{Kl}(\mathbf{T})$ with $|\mathrm{Kl}(\mathbf{T})| = |\mathcal{C}|$ and $\mathrm{Kl}(\mathbf{T})(X, Y) = \mathcal{C}(X, TY)$. We make free use of the well-known fact that for a co-Cartesian $\mathcal{C}$ and a monad $\mathbf{T}$ on $\mathcal{C}$, the Kleisli category $\mathrm{Kl}(\mathbf{T})$ is again co-Cartesian with the coproduct co-spans $X \xrightarrow{\eta\,\mathsf{inl}} T(X+Y) \xleftarrow{\eta\,\mathsf{inr}} Y$ and $[(T\,\mathsf{inl})f, (T\,\mathsf{inr})g] \colon X + Y \to T(X' + Y')$ being the coproduct of morphisms $f \colon X \to TX'$ and $g \colon Y \to TY'$.

Unless stated otherwise, all diagrams we present are supposed to commute.

3 Free Extensions

We recall the following standard notion, see e.g. [3, Section 7.7].

▶ **Definition 1** (Bimodules). *For categories $\mathcal{C}$ and $\mathcal{D}$, a* bimodule $\mathcal{O} \colon \mathcal{C} \nrightarrow \mathcal{D}$ *consists of the following data:*

- *a family of sets $(\mathcal{O}(X, \underline{Y}))_{X \in |\mathcal{C}|, \underline{Y} \in |\mathcal{D}|}$, where $g \in \mathcal{O}(X, \underline{Y})$ is called an $\mathcal{O}$-*morphism *$g \colon X \to \underline{Y}$;*

- *each $g\colon X \to \underline{Y}$ can be composed with a $\mathcal{C}$-map $f\colon X' \to X$ or $\mathcal{D}$-map $h\colon \underline{Y} \to \underline{Y'}$.*

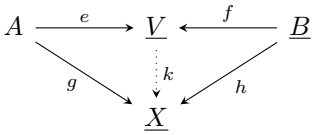
For $g\colon X \to \underline{Y}$, $f' : X'' \to X'$, $f\colon X' \to X$, $h'\colon \underline{Y'} \to \underline{Y''}$, $h\colon \underline{Y} \to \underline{Y'}$ we must have the following:

$$g\,\mathsf{id}_X = g \qquad (h'\,h)g = h'\,(h\,g) \qquad h\,(g\,f) = (h\,g)\,f$$
$$\mathsf{id}_{\underline{Y}}\,g = g \qquad g\,(f\,f') = (g\,f)\,f'$$

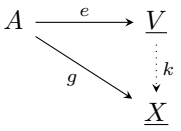
For example: the bimodule **Set** $\to$ **Ring** in which $\mathcal{O}(X, \underline{Y})$ is the set of functions from the set X to the ring $\underline{Y}$. This bimodule can be seen as arising from the forgetful functor **Ring** $\to$ **Set**.

Bimodules $\mathcal{C} \nrightarrow \mathcal{D}$ correspond to functors $\mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to$ **Set**. They are also called *distributors* or *profunctors* (but some authors reverse the direction). For the rest of the section, let $\mathcal{O}\colon \mathcal{C} \nrightarrow \mathcal{D}$ be a bimodule.

▶ **Definition 2** (Free Extensions). *Let $A \in |\mathcal{C}|$ and $\underline{B} \in |\mathcal{D}|$. A* free extension *of $\underline{B}$ by A consists of $\underline{V} \in |\mathcal{D}|$ and $e\colon A \to \underline{V}$ and $f\colon \underline{B} \to \underline{V}$, such that, for all $\underline{X} \in |\mathcal{D}|$ and $g\colon A \to \underline{X}$ and $h\colon \underline{B} \to \underline{X}$, there is a unique $k\colon \underline{V} \to \underline{X}$ such that*



▶ **Definition 3** (Free Objects). *Let $A \in |\mathcal{C}|$. A* free object *on A consists of $\underline{V} \in |\mathcal{D}|$ and $e\colon A \to \underline{V}$, such that, for all $\underline{X} \in |\mathcal{D}|$ and $g\colon A \to \underline{X}$, there is a unique $k\colon \underline{V} \to \underline{X}$ such that*



▶ **Proposition 4.** *Let $0_{\mathcal{D}}$ be an initial object in $\mathcal{D}$. For any $A \in |\mathcal{C}|$, a free object on A corresponds to a free extension of $0_{\mathcal{D}}$ by A. The bijection sends $(\underline{V}, e)$ to*

$$A \xrightarrow{\ e\ } \underline{V} \xleftarrow{\ [\,]\ } 0_{\mathcal{D}}.$$

▶ **Proposition 5.** *Let $A \in |\mathcal{C}|$ and $\underline{B} \in |\mathcal{D}|$. Let $(\underline{W}, d)$ be a free object on A. Then a coproduct of $\underline{W}$ and $\underline{B}$ corresponds to a free extension of $\underline{B}$ by A. The bijection sends $(\underline{V}, e, f)$ to*

$$A \xrightarrow{\ d\ } \underline{W} \xrightarrow{\ e\ } \underline{V} \xleftarrow{\ f\ } \underline{B}.$$

4 Guardedness on Monads

In this section, let $\mathcal{K}$ be a co-Cartesian category. The typical example is $\mathcal{K} = \mathrm{Kl}(\mathbf{T})$, where $\mathbf{T}$ is a monad on a co-Cartesian category $\mathcal{C}$.

4.1 Guardedness Predicates

The following notion is slightly adapted from [10].

▶ **Definition 6** (Guardedness, Guarded Monads)**.** *A* guardedness predicate *on $\mathcal{K}$ provides for all objects X, Y, Z a subset $\mathcal{K}^\bullet(X,Y,Z) \subseteq \mathcal{K}(X, Y+Z)$. We write $f\colon X \to Y \rangle Z$ for $f \in \mathcal{K}^\bullet(X,Y,Z)$ and say that f is* guarded *(in the right summand). The following conditions are required:*

$$\textbf{(trv)}\quad \frac{f\colon X \to Y}{\mathsf{inl} f : X \to Y \rangle Z} \qquad \textbf{(par)}\quad \frac{f\colon X \to V \rangle W \qquad g\colon Y \to V \rangle W}{[f,g] : X + Y \to V \rangle W}$$

$$\textbf{(cmp)}\quad \frac{f\colon X \to Y \rangle Z \qquad g\colon Y \to V \rangle W \qquad h\colon Z \to V + W}{[g,h]f : X \to V \rangle W}$$

A category equipped with a guardedness predicate is called guarded category*. A monad $\mathbf{T}$ on $\mathcal{C}$ is a* guarded monad *if $\mathcal{K} = \mathrm{Kl}(\mathbf{T})$ is a guarded category under the coproducts inherited from $\mathcal{C}$.*

We write "let $f\colon X \to Y \rangle Z$" as an abbreviation for "let f be a map $X \to Y + Z$ be a map such that $f\colon X \to Y \rangle Z$".

Intuitively, a morphism $f\colon X \to Y \rangle Z$ represents a program flow with inputs in X and outputs in Y and in Z, where the latter part of the output is guarded in the sense that every portion of the program flow from X to Z runs through a guard. The notion of guard here is implicit and depends on the specific model. The axioms of guardedness abstractly capture properties of guards: **(trv)** states that if all the output goes to Y then $f\colon X \to Y + Z$ is (vacuously) guarded in Z; **(par)** states that guardedness jointly depends on all inputs; finally, **(cmp)** states that if the program flow branches then every branch leading to the guarded output must hit a guard at least once, specifically, $h\colon Z \to V + W$ need not be guarded in W, because h receives the input from f, which ensures guarded already.

The distinction between Definition 6 and the corresponding definition in [10] is precisely determined by the choice of the notion of coproduct: in op. cit. coproducts are treated up to isomorphisms, while here we work with selected coproducts. The original axiomatization of guardedness additionally involved a *weakening rule*, which turned out to be derivable from the above three [8]. Let us summarize this and other consequences of the axioms. We will need the following convention.

▶ **Notation 7.** Let us use the notation $f\colon X \to Y \rangle Y_1 \rangle \ldots \rangle Y_n$, for $f\colon X \to (\ldots(Y + Y_1) + \ldots) + Y_n$ meaning that $\sigma f\colon X \to Y \rangle Y_1 + \ldots + Y_n$ where σ is the obvious associativity isomorphism $(\ldots(Y + Y_1) + \ldots) + Y_n \to Y + (Y_1 + (\ldots + Y_n)\ldots)$.

▶ **Proposition 8.** *Let $\mathcal{K}$ be a guarded category.*

1. *For all objects $V, W \in |\mathcal{K}|$, we have $[\,]\colon 0 \to V \rangle W$.*
2. *Let $f\colon X \to Y \rangle Z$. For $u\colon X' \to X$ and $g\colon Y \to Y'$ and $h\colon Z \to Z'$ we have $(g+h)fu\colon X' \to Y' \rangle Z'$.*
3. *(Weakening) If $f\colon X \to Y \rangle Z \rangle W$ then $f\colon X \to Y + Z \rangle W$.*

It is often useful to speak of guardedness in particular summands:

- we say that $f\colon X \to Y + Z$ is inr-guarded if $f\colon X \to Y \rangle Z$;
- we say that $f\colon X \to Y + Z$ is inl-guarded if $X \xrightarrow{f} Y + Z \cong Z + Y$ is inr-guarded;
- we say that $f\colon X \to Y$ is id-guarded if $X \xrightarrow{f} Y \cong 0 + Y$ is inr-guarded.

Two guardedness predicates are especially important.

▶ **Proposition 9** (Greatest and Least Guardedness Predicates)**.**

1. *The greatest guardedness predicate on $\mathcal{K}$ says that, for* every *map $f\colon X \to Y + Z$, we have $f\colon X \to Y \rangle Z$.*
2. *The least guardedness predicate on $\mathcal{K}$ says that, for $f\colon X \to Y + Z$, $f\colon X \to Y \rangle Z$ iff there is a map $g\colon X \to Y$ such that f factors as $X \xrightarrow{g} Y \xrightarrow{\mathsf{inl}} Y + Z$ (such g need not be unique, since it does not follow from our running premises that coproduct injections are monic).*

We say that $\mathcal{K}$ is *totally guarded* when equipped with the largest guardedness predicate, and *vacuously guarded* when equipped with the smallest.

▶ **Example 10.** Here are some examples of guardedness predicates for $\mathcal{K} = \mathrm{Kl}(\mathbf{T})$ with $\mathbf{T}$ being a monad on **Set**.

1. Let $\mathbf{T}$ be the following monad: $T\emptyset = \emptyset$ and $TX = 1$ if $X \neq \emptyset$, under vacuous guardedness. Now, the unique morphism $1 \to T(1+1) = 1$ is inl-guarded and inr-guarded, because it factors through $1 = T1 \xrightarrow{T\,\mathsf{inr}} T(1+1) = 1$ and through $1 = T1 \xrightarrow{T\,\mathsf{inl}} T(1+1) = 1$. But $1 \to T(1+1) = 1$ does not factor through $\emptyset = T\emptyset \xrightarrow{T[]} T(1+1) = 1$, and hence it is not id-guarded. This example show that guardedness in two summands does not necessarily imply guardedness in their union.
2. Let $\mathcal{P}^+$ be the non-empty powerset monad. For $f\colon X \to \mathcal{P}^+(Y + Z)$, say $f\colon X \to Y \rangle Z$ when for every $x \in X$, the set $f(x)$ contains at least one element of the form $\mathsf{inl}\, y$.
3. Let $\mathcal{D}^+$ be the *countable probability distribution monad*:

$$\mathcal{D}^+X = \left\{ d : X \to [0,1] \mid \sum d = 1 \right\}.$$

   We put $f\colon X \to Y \rangle Z$ if for every $x \in X$, $f(x)(\mathsf{inl}\, y) > 0$ for at least one $y \in Y$.
4. For a set A, let $TX = A^\star \times X$ be a *writer monad* whose monad structure is induced by the monoid structure of $A^\star$. For $f\colon X \to A^\star \times (Y + Z)$, say $f\colon X \to Y \rangle Z$ when, for every $x \in X$, if $f(x) = (m, \mathsf{inr}\, z)$ then $m \neq \varepsilon$.
5. Following Section 1.1, let A be again an arbitrary set and let $TX = \mathcal{P}^+(A^\star \times X + A^\omega)$. This yields a monad for nondeterministic programs that print characters in A, giving semantics that records the (successful) finite and infinite traces. The monad structure is obtained from the fact that $A^\star$ is a monoid and A^ω is a left $A^\star$-module. For $f\colon X \to \mathcal{P}^+(A^\star \times (Y + Z) + A^\omega)$, say $f\colon X \to Y \rangle Z$ when, for every $x \in X$, if $(m, \mathsf{inr}\, z) \in f(x)$ then $m \neq \varepsilon$. Intuitively, as in the previous example, a program denoting f is prohibited from returning a value through Z without first printing a character.

▶ **Definition 11** (Guarded Natural Transformations and Monad Morphisms)**.**

1. *Let $\mathbf{T}$ and $\mathbf{S}$ be guarded monads on $\mathcal{C}$. A monad morphism $\rho\colon \mathbf{T} \to \mathbf{S}$ is* guarded *when the functor $\mathrm{Kl}(\rho)\colon \mathrm{Kl}(\mathbf{T}) \to \mathrm{Kl}(\mathbf{S})$ preserves guardedness. Explicitly: for $f\colon X \to T(Y + Z)$, if $f\colon X \to Y \rangle Z$ in $\mathrm{Kl}(\mathbf{T})$ then $X \xrightarrow{f} T(Y + Z) \xrightarrow{\rho_{Y+Z}} S(Y + Z)$ is guarded $X \to Y \rangle Z$ in $\mathrm{Kl}(\mathbf{S})$.*
2. *Let H be an endofunctor and $\mathbf{T}$ a guarded monad on $\mathcal{C}$. A natural transformation $\sigma\colon H \to T$ is* guarded *when for all $X \in |\mathcal{C}|$, $\sigma_X\colon HX \to TX$ is id-guarded.*

4.2 Guarded Iteration

We now consider when guarded morphisms can be iterated in the sense of Section 1.1. The most straightforward case is the following:

▶ **Definition 12** (Guarded Iterative Categories). *$\mathcal{K}$ is* guarded iterative *if every $f\colon X \to Y \rangle X$ has a unique fixpoint $f^\dagger\colon X \to Y$ of the map $[\mathsf{id}, -]\, f\colon \mathcal{K}(X,Y) \to \mathcal{K}(X,Y)$.*

▶ **Lemma 13.** *In any guarded category, if $f\colon X \to Y \rangle Z \rangle X$ and $g\colon X \to Y$ is a fixpoint of $[\mathsf{id}, -]\, f$ then $g\colon X \to Y \rangle Z$.*

▶ **Definition 14** (Conway Iteration). *A* guarded Conway (iteration) operator *on $\mathcal{K}$ associates to each $f\colon X \to Y \rangle X$ a fixpoint $f^\dagger\colon X \to Y$ of the map $[\mathsf{id}, -]\, f$, satisfying the following principles:*

- naturality: *for $f\colon X \to Y \rangle X$ and $g\colon Y \to Z$ we have $((g + \mathsf{id})f)^\dagger = gf^\dagger$;*
- dinaturality: $([\mathsf{inl}, h]\, g)^\dagger = [\mathsf{id}, ([\mathsf{inl}, g]\, h)^\dagger]\, g$ *for $g : X \to Y \rangle Z$ and $h\colon Z \to Y \rangle X$ or $g : X \to Y + Z$ and $h\colon Z \to Y \rangle X$;*
- codiagonal: $([\mathsf{id}, \mathsf{inr}]\, f)^\dagger = f^{\dagger\dagger}$ *for $f : X \to Y \rangle X \rangle X$.*

Note that in the codiagonal equation, $f^{\dagger\dagger}$ must exist by Lemma 13.

▶ Remark 15. Guarded Conway operators are direct generalizations of standard (total) Conway operators [2, 22], which arise under the total notion of guardedness. It was observed by Hyland and Hasegawa [12, 11] that Conway operators are equivalent to monoidal trace operators under $\otimes = +$ (modulo the duality of $+$ and $\times$). The connection between Conway operators and traces extends to a connection between guarded Conway operators and *guarded traces* [8]. In the total case, it is known that the axioms of Conway operators are incomplete wrt nontrivial models of iteration, e.g. the category of pointed complete partial orders [22]. This led Bloom and Ésik to completing the axiomatization of iteration by an infinite set of axioms called *commutative identities* [2]. These identities are instance of a single versatile quasi-equational *uniformity* principle, which holds true in all non-pathological models.

Let $J\colon \mathcal{C} \to \mathcal{K}$ be a functor, where $\mathcal{C}$ and $\mathcal{K}$ are guarded and have the same objects, and J is identity-on-objects and strictly preserves co-Cartesian structure.

▶ **Definition 16** (Uniformity). *A guarded Conway operator $-^\dagger$ on $\mathcal{K}$ is* uniform *(wrt J) when for $\mathcal{K}$-maps $f\colon X \to Y \rangle X$ and $g\colon Z \to Y \rangle Z$ and $\mathcal{C}$-map $h\colon Z \to X$,*

$$\begin{array}{ccc} Z & \xrightarrow{g} & Y + Z \\ {\scriptstyle Jh}\downarrow & & \downarrow{\scriptstyle Y + Jh} \\ X & \xrightarrow{f} & Y + X \end{array} \quad \Rightarrow \quad \begin{array}{ccc} Z & \xrightarrow{g^\dagger} & Y \\ {\scriptstyle Jh}\downarrow & \nearrow_{f^\dagger} & \\ X & & \end{array}$$

▶ **Proposition 17.** *[10] An operation sending every $f \in \mathcal{K}^\bullet(X,Y,Z)$ to a fixpoint $f^\dagger \in \mathcal{K}(X,Y)$ is guarded Conway uniform iff it satisfies naturality, codiagonal and uniformity. In other words, dinaturality is derivable.*

▶ **Proposition 18.** *Let $\mathcal{K}$ be guarded iterative. Then $f \mapsto f^\dagger$ is a guarded Conway operator and uniform wrt $\mathrm{Id}_\mathcal{K}$.*

Proof. Except for uniformity wrt $\mathrm{Id}_\mathcal{K}$, the proof is in [10, Theorem 17]. Let us verify the missing case of uniformity. Suppose that $f\,(Jh) = J(\mathsf{id} + h)\, g$ for suitable f, g and h. Now, $[\mathsf{id}, f^\dagger(Jh)]\, g = [\mathsf{id}, f^\dagger]\, J(\mathsf{id} + h)g = [\mathsf{id}, f^\dagger]\, fJh$, meaning that $f^\dagger(Jh)$ satisfies the fixpoint equation for $g^\dagger$. Therefore, $f^\dagger(Jh) = g^\dagger$. ◀

▶ **Definition 19.** *Let $\mathbf{T}$ be a guarded monad, i.e. a monad with a guardedness predicate on $\mathrm{Kl}(\mathbf{T})$. We say that $\mathbf{T}$ is*

1. *a* guarded iterative monad *if $\mathrm{Kl}(\mathbf{T})$ is a guarded iterative category;*
2. *a* guarded Elgot monad *if $\mathrm{Kl}(\mathbf{T})$ has a guarded Conway operator $f \mapsto f^\dagger$, which is uniform wrt the obvious functor $\mathcal{C} \to \mathrm{Kl}(\mathbf{T})$;*

3. *an* Elgot monad *when it is totally guarded and a guarded Elgot monad.*

Note that for an Elgot monad $\mathbf{T}$, $T0$ must always be inhabited because $\mathbf{T}$ supports (unproductive) divergence $\bot = (\eta\,\mathsf{inr}\colon 1 \to T(0+1))^\dagger$.

▶ **Example 20.** Let us revisit Example 10.

1. Every monad under vacuous guardedness can be equipped with an iteration operator and is thus guarded iterative. Concretely, since $f\colon X \to Y \rangle X$ implies that

$$f = \big(X \xrightarrow{g} TY \xrightarrow{T\,\mathsf{inl}} T(Y+X)\big)$$

for a suitable g, $f^\dagger = [\eta, f^\dagger]^\star\,(T\,\mathsf{inl})g = g$. For Example 10 (1), therefore $f^\dagger = !\colon X \to TY = 1$ if $Y \neq \emptyset$ and $f^\dagger = [\,] : \emptyset \to \emptyset$ if $Y = \emptyset$, and thus $X = \emptyset$.

2. The powerset monad $\mathcal{P}$ is Elgot, because its Kleisli category (the category of relations) is enriched over complete partial orders, and hence supports $f^\dagger$ as a least fixpoint of $[\eta, -]^\star f$. This is inherited by $\mathcal{P}^+$ by restriction along the inclusion $\mathcal{P}^+ \hookrightarrow \mathcal{P}$. Explicitly, in the Kleisli category of $\mathcal{P}^+$, for a guarded map $f\colon X \to Y \rangle X$, the map $f^\dagger\colon X \to Y$ sends x to the set

$$\big\{y \in Y \mid \exists n \in \mathbb{N}, (x_0, \ldots, x_n) \in X^{n+1}.\, x = x_0 \wedge f(x_0) \ni \mathsf{inr}\, x_1 \wedge \ldots \wedge f(x_n) \ni \mathsf{inl}\, y\big\}.$$

In this style of semantics we thus do not register the possibility of divergence i.e. whether there is a sequence $(x_0, x_1, \ldots) \in X^\omega$ such that $\forall i \in \mathbb{N}.\, f(x_i) \ni \mathsf{inr}\, x_{i+1}$. As a result, $\mathcal{P}^+$ is guarded Elgot.

3. Unlike its cousin, the *countable subdistribution monad* $\mathcal{D}X = \big\{d : X \to [0,1] \mid \sum d \leq 1\big\}$, $\mathcal{D}^+$ is not Elgot (because $\mathcal{D}^+0 = 0$). However, as in the previous clause, it is guarded Elgot. Specifically, we obtain a guarded Conway operator for $\mathcal{D}^+$ by first restricting from the corresponding total guarded iteration operator for $\mathcal{D}$, calculated as a least fixed point, and then normalizing (guardedness ensures it is not zero) to obtain a distribution. Thus we do not record the probability of divergence. To see the need for normalization, consider the guarded Kleisli map $f\colon \mathbb{N} \to 1 \rangle \mathbb{N}$ where $f(n)$ gives $\mathsf{inl}\star$ with probability

$$\frac{1}{2^n+2} = \left(\frac{1}{2^{n+1}}\right) / \left(\frac{1}{2} + \frac{1}{2^n}\right)$$

and $\mathsf{inr}(n+1)$ with probability

$$1 - \frac{1}{2^n+2} = \frac{2^n+1}{2^n+2} = \left(\frac{1}{2} + \frac{1}{2^{n+1}}\right) / \left(\frac{1}{2} + \frac{1}{2^n}\right).$$

Iterating f from 0 gives the probability $1/2^{n+2}$ of the transition sequence

$$0 \to 1 \to \cdots \to n \to \star$$

So the probability of eventually reaching $\star$ is $1/2$.

4. The writer monad $TX = A^\star \times X$ does not support guarded iteration for the guardedness predicate defined in Example 10 (4) and for non-trivial A. For example, no $x : 1 \to T1 \cong A^\star$ satisfies the fixpoint equation $x = ax$ for any $a \in A$. This can be remedied by extending TX to $A^\star \times X + M$ were M is an inhabited left $A^\star$-module, e.g. $M = 1$ (the initial one), or $M = A^\omega$ (the final one).

5. The monad $TX = \mathcal{P}^+(A^\star \times X + A^\omega)$ for finite and infinite traces from Example 10 (5) supports the following iteration operator. For a guarded Kleisli map $f\colon X \to Y \rangle X$, the

map $f^\dagger \colon X \to TY$ sends x to the following set:

$$\begin{aligned}
&\big\{\mathsf{inl}(m_0 + \cdots + m_{n-1} + m, y) \mid \exists n \in \mathbb{N}, (x_0, \ldots, x_n) \in X^{n+1}.\\
&\qquad x_0 = x\\
&\qquad \wedge f(x_0) \ni \mathsf{inl}(m_0, \mathsf{inr}\, x_1) \wedge \ldots \wedge f(x_{n-1}) \ni \mathsf{inl}(m_{n-1}, \mathsf{inr}\, x_n)\\
&\qquad \wedge f(x_n) \ni \mathsf{inl}(m, \mathsf{inl}\, y)\big\}\\
\cup &\big\{\mathsf{inr}(m_0 + \cdots + m_{n-1} + m) \mid \exists n \in \mathbb{N}, (x_0, \ldots, x_n) \in X^{n+1}.\\
&\qquad x_0 = x\\
&\qquad \wedge f(x_0) \ni \mathsf{inl}(m_0, \mathsf{inr}\, x_1) \wedge \ldots \wedge f(x_{n-1}) \ni \mathsf{inl}(m_{n-1}, \mathsf{inr}\, x_n)\\
&\qquad \wedge f(x_n) \ni \mathsf{inr}\, m\big\}\\
\cup &\big\{\mathsf{inr}(m_0 + m_1 + \cdots) \mid \exists (x_0, \ldots) \in X^\omega.\\
&\qquad x_0 = x \wedge \forall i \in \mathbb{N}.\, f(x_i) \ni \mathsf{inl}(m_i, \mathsf{inr}\, x_{i+1})\big\}.
\end{aligned}$$

This captures three possible scenarios (separated by the $\cup$ operator):

- the fixpoint $f^\dagger(x)$ is unfolded n times resulting in an output of $y \in Y$; the actions $m_0, \ldots, m_{n-1}, m \in A^\star$ are collected along the run and concatenated;
- the fixpoint $f^\dagger(x)$ is unfolded n times and then hits an infinite trace $m \in A^\omega$; as a result, $f^\dagger(x)$ does not yield a value from Y, but it yields an infinite trace $m_0 + \ldots + m_{n-1} + m \in A^\omega$ where $m_0, \ldots, m_{n-1} \in A^\star$ are collected along the run;
- the fixpoint $f^\dagger(x)$ is unfolded infinitely many times without ever reaching Y; this yields an infinite trace $m_0 + m_1 + \ldots \in A^\omega$ computed by concatenating the traces $m_i \in A^\star$, which are collected along the run. The guardedness assumption on f is crucial here, because it ensure that each m_i is non-empty and hence the above infinite sum does indeed produce an infinite trace.

The resulting iteration operator is properly partial, and is computed neither as a least fixpoint nor as a unique fixpoint, even though the guardedness relation we postulate is the one standardly used in process algebra and guaranteeing uniqueness of fixpoint under strong bisimilarity [17]. The separating example is $x = ax + 1$, which has besides the canonical solution $x = a^\star + a^\omega$ the solution $x = a^\star$, ignoring the infinite trace.

5 The Coinductive Resumption Monad

In this section, we present our main technical contribution, stating that guarded Elgotness extends along the coalgebraic resumption monad transformer. It proves to be technically more advantageous to work more generally with *parametrized guarded Elgot monads*, which extend Uustalu's *parametrized monads* [23].

▶ **Definition 21** (Parametrized Guarded Elgot Monads). *A* parametrized guarded Elgot monad *is a functor from a co-Cartesian category $\mathcal{C}$ to the category of guarded Elgot monads over $\mathcal{C}$. Equivalently (by uncurrying), a parametrized guarded Elgot monad is a bifunctor $\# : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, such that each $- \# W$ is a guarded Elgot monad, and for every $f \colon W \to W'$, $- \# f$ is a guarded Elgot monad morphism.*

Since every monad is a guarded Elgot monad under vacuous guardedness, parametrized guarded Elgot monads include all parametrized monads.

The main example of a parametrized guarded Elgot monad is as follows.

▶ **Example 22.** Given a guarded Elgot monad $\mathbf{T}$ and an endofunctor H on the same co-Cartesian category $\mathcal{C}$, $X \# Y = T(X + HY)$ defines a parametrized guarded Elgot monad, with

the guardedness predicate defined as follows: for $f\colon X \to T((Y+Z)+HW)$, $f\colon X \to Y \rangle Z$ in $\mathrm{Kl}(- \# W)$ iff

$$X \xrightarrow{f} T((Y+Z)+HW) \cong T((Y+HW)+Z) \qquad \text{is} \qquad \mathsf{inr}\text{-guarded in } \mathrm{Kl}(\mathbf{T}).$$

We use the same Kleisli style notation for parametrized guarded Elgot monads as for the non-parametrized ones. Let us record the identities, directly implied by Definition 21, and which we use in the subsequent calculations.

$$\begin{aligned}
(\eta_{X,Y} : X \to Z \# W)^\star &= \mathsf{id}_{X\#W} : X \# W \to X \# W,\\
(f\colon Y \to Z \# W)^\star(\eta_{X,Y} : X \to X \# W) &= f\colon X \to Z \# W,\\
(f\colon Y \to Z \# W)^\star(g\colon X \to Y \# W)^\star &= (f^\star g)^\star : X \# W \to Z \# W,\\
(X \# (h\colon W \to W'))\,(\eta_{X,W} : X \to X \# W) &= \eta_{X,W'} : X \to X \# W',\\
(Y \# (h\colon W \to W'))\,(f\colon X \to Y \# W)^\star &= \\
&\qquad ((Y \# h)f)^\star(Y \# h) : X \# W \to Y \# W',\\
(Y \# (h\colon W \to W'))\,(f\colon X \to (Y+X) \# W)^\dagger &= ((Y \# h)\,f)^\dagger : X \to Y \# W'.
\end{aligned}$$

The first three equations here are the monad laws for $- \# W$ and the last three equations express the fact that $- \# h$ is a guarded Elgot monad morphism.

For the rest of the section we fix a parametrized guarded Elgot monad $\#$ and assume that the final coalgebras $\Gamma_\# X = \nu\gamma.\, X \# \gamma$ exist for all $X \in |\mathcal{C}|$. The following properties of $\Gamma_\#$ are previously shown by Uustalu [23].

▶ **Proposition 23.**

1. *For every $f\colon X \to \Gamma_\# Y$ there is a unique $f^\bullet\colon \Gamma_\# X \to \Gamma_\# Y$ such that*

$$\begin{array}{ccc}
\Gamma_\# X & \xrightarrow{\mathsf{out}} & X \# \Gamma_\# X \\
{\scriptstyle f^\bullet}\downarrow & & \downarrow{\scriptstyle ((\mathsf{out}\, f)\# f^\bullet)^\star} \\
\Gamma_\# Y & \xrightarrow{\mathsf{out}} & Y \# \Gamma_\# Y
\end{array}$$

2. *given $f\colon X \to Y \# \Gamma_\#(X+Y)$, there is unique $g\colon X \to \Gamma_\# Y$, such that*

$$\begin{array}{ccc}
X & \xrightarrow{f} & Y \# \Gamma_\#(Y+X) \\
{\scriptstyle g}\downarrow & & \downarrow{\scriptstyle Y \# [\eta^\nu,\, g]^\bullet} \\
\Gamma_\# Y & \xrightarrow{\mathsf{out}} & Y \# \Gamma_\# Y
\end{array}$$

3. *$\Gamma_\#$ forms a monad, whose unit $\eta^\#$ at X is*

$$X \xrightarrow{\eta\,\mathsf{inl}} X \# \Gamma_\# X \xrightarrow{\mathsf{out}^{-1}} \Gamma_\# X.$$

The Kleisli extension of $f\colon X \to \Gamma_\# Y$ is $f^\bullet$.

5.1 Transferring Guarded Elgotness

We proceed to transfer iteration from $\#$ to $\Gamma_\#$.

▶ **Definition 24.** *$\Gamma_\#$ is guarded as follows: $f\colon X \to Y \rangle Z$ when the composite*

$$X \xrightarrow{f} \Gamma_\#(Y+Z) \xrightarrow{\mathsf{out}} (Y+Z) \# \Gamma_\#(Y+Z)$$

is inr*-guarded.*

See [10] for a proof that this constitutes a guardedness predicate.

Note that when $\#$ is totally guarded then so is $F_\#$. Using Definition 24, for every $f\colon X \to F_\#(Y+X)$, we define $\diamondsuit f = (\mathsf{out}\, f)^\dagger : X \to Y \# F_\#(Y+X)$. The idea of this operator is as follows. Computing the iteration of $f\colon X \to F_\#(Y+X)$ w.r.t. $F_\#$ amounts to forming $\mathsf{out}\, f\colon X \to (Y+X) \# F_\#(Y+X)$ first, which reveals two occurrences of X that must be iterated away. The first one occurs at the guarded position of the parametrized monad, and hence we can eliminate it by applying the iteration operator of $- \# F_\#(Y+X)$ – this is precisely the task of $\diamondsuit$. The remaining second position of X occurs under $F_\#$, and can be eliminated by using the finality property of the latter.

▶ **Theorem 25.** *$F_\#$ is a guarded Elgot monad with the iteration operator $(-)^\ddagger$ characterized as follows: for $f\colon X \to Y \rangle X$, $f^\ddagger\colon X \to F_\# Y$ is the unique morphism satifsying*

$$\begin{array}{ccc} X & \xrightarrow{\diamondsuit f} & Y \# F_\#(Y+X) \\ {\scriptstyle f^\ddagger}\downarrow & & \downarrow{\scriptstyle Y \# [\eta^\nu, f^\ddagger]^\star} \\ F_\# Y & \xrightarrow{\mathsf{out}} & Y \# F_\# Y \end{array}$$

Proof. Let us verify the relevant laws. Recall that by Proposition 17, we need not verify dinaturality.

- *fixpoint* is already shown in [10];
- *naturality:* given $f\colon X \to Y \rangle X$, $g\colon Y \to F_\# Z$, let us denote by h the morphism $[(F_\# \mathsf{inl})\, g, \eta^\nu\, \mathsf{inr}] : Y + X \to Z \rangle X$, First we show that

$$\diamondsuit(h^\star f) = ((Z \# F_\# \mathsf{inl})\, \mathsf{out}\, g)^\star (Y \# h^\star) \diamondsuit f. \tag{1}$$

Indeed,

$$\begin{aligned} \diamondsuit(h^\star f) &= (\mathsf{out}\ h^\star f)^\dagger && /\!/ \text{ definition of } \diamondsuit \\ &= ((\mathsf{out}\, h)^\star (Y \# h^\star)\ \mathsf{out}\ f)^\dagger \\ &= ((Z \# F_\# \mathsf{inl})\, \mathsf{out}\, g)^\star ((Y \# h^\star)\, \mathsf{out}\, f)^\dagger && /\!/ \text{ naturality of } (-)^\dagger \\ &= ((Z \# F_\# \mathsf{inl})\, \mathsf{out}\, g)^\star (Y \# h^\star) \diamondsuit f. \end{aligned}$$

Then

$$\begin{aligned} \mathsf{out}\, g^\star f^\ddagger &= (\mathsf{out}\, g)^\star (Y \# g^\star)\, (Y \# [\eta^\nu, f^\ddagger]^\star) \diamondsuit f && /\!/ \text{ definition of } (-)^\ddagger \\ &= (\mathsf{out}\, g)^\star (Y \# (g^\star [\eta^\nu, f^\ddagger]^\star)) \diamondsuit f \\ &= (\mathsf{out}\, g)^\star \big(Y \# [\eta^\nu, g^\star f^\ddagger]^\star [(F_\# \mathsf{inl})\, g, \eta^\nu\, \mathsf{inr}]^\star\big) \diamondsuit f \\ &= (\mathsf{out}\, g)^\star \big(Y \# [\eta^\nu, g^\star f^\ddagger]^\star h^\star\big) \diamondsuit f \\ &= (Z \# [\eta^\nu, g^\star f^\ddagger]^\star)\, ((Z \# F_\# \mathsf{inl})\, \mathsf{out}\, g)^\star (Y \# h^\star) \diamondsuit f && /\!/ \ (1) \\ &= (Z \# [\eta^\nu, g^\star f^\ddagger]^\star) \diamondsuit (h^\star f). \end{aligned}$$

This entails the requisite equality $(h^\star f)^\ddagger = g^\star f^\ddagger$, by the uniqueness property of $(h^\star f)^\ddagger$.

- *codiagonal:* let $f : X \to Y \rangle X \rangle X$. It suffices to check that

$$\mathsf{out}\, f^{\ddagger\ddagger} = (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\star) \diamondsuit (F_\# [\mathsf{id}, \mathsf{inr}]\, f)$$

The proof runs as follows:

$$\begin{aligned}
\mathsf{out}\, f^{\ddagger\ddagger} &= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet) \diamondsuit f^\ddagger && /\!\!/ \text{ definition of } (-)^\ddagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\, (\mathsf{out}\, f^\ddagger)^\dagger && /\!\!/ \text{ definition of } \diamondsuit \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\, (((Y + X) \# [\eta^\nu, f^\ddagger]^\bullet) \diamondsuit f)^\dagger && /\!\!/ \text{ definition of } (-)^\ddagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet [\eta^\nu, f^\ddagger]^\bullet)\, (\mathsf{out}\, f)^{\dagger\dagger} \\
&= (Y \# [[\eta^\nu, f^{\ddagger\ddagger}], [\eta^\nu, f^{\ddagger\ddagger}]^\bullet f^\ddagger]^\bullet)\, (\mathsf{out}\, f)^{\dagger\dagger} \\
&= (Y \# [[\eta^\nu, f^{\ddagger\ddagger}], f^\ddagger]^\bullet)\, (\mathsf{out}\, f)^{\dagger\dagger} && /\!\!/ \text{ fixpoint for } (-)^\ddagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet F_\# [\mathsf{id}, \mathsf{inr}])\, (\mathsf{out}\, f)^{\dagger\dagger} \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\, (Y \# F_\# [\mathsf{id}, \mathsf{inr}]) \\
&\qquad (([\mathsf{id}, \mathsf{inr}] \# F_\# ((Y + X) + X))\, \mathsf{out}\, f)^\dagger && /\!\!/ \text{ codiagonal for } (-)^\dagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\, (([\mathsf{id}, \mathsf{inr}] \# F_\# [\mathsf{id}, \mathsf{inr}])\, \mathsf{out}\, f)^\dagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\, (\mathsf{out}\ F_\# [\mathsf{id}, \mathsf{inr}] f)^\dagger \\
&= (Y \# [\eta^\nu, f^{\ddagger\ddagger}]^\bullet)\ \diamondsuit (F_\# [\mathsf{id}, \mathsf{inr}] f) && /\!\!/ \text{ definition of } \diamondsuit
\end{aligned}$$

- *uniformity:* assume $f \colon X \to Y \rangle X$, $g \colon Z \to Y \rangle Z$, $h \colon Z \to X$ and $f\, h = F_\# (\mathsf{id} + h)\, g$. The latter entails $(\mathsf{out}\, f)\, h = ((\mathsf{id} + h) \# F_\# (\mathsf{id} + h))\ \mathsf{out}\, g$, hence, by uniformity of $(-)^\dagger$,

$$(\mathsf{out}\, f)^\dagger\, h = ((Y \# F_\# (\mathsf{id} + h))\ \mathsf{out}\, g)^\dagger. \tag{2}$$

We then have

$$\begin{aligned}
\mathsf{out}\, f^\ddagger\, h &= (Y \# [\eta^\nu, f^\ddagger]^\bullet)\, (\diamondsuit f)\, h && /\!\!/ \text{ definition of } (-)^\ddagger \\
&= (Y \# [\eta^\nu, f^\ddagger]^\bullet)\, (\mathsf{out}\, f)^\dagger\, h && /\!\!/ \text{ definition of } \diamondsuit \\
&= (Y \# [\eta^\nu, f^\ddagger]^\bullet)\, ((Y \# F_\# (\mathsf{id} + h))\ \mathsf{out}\, g)^\dagger && /\!\!/ \ (2) \\
&= (Y \# [\eta^\nu, f^\ddagger h]^\bullet)\, (\mathsf{out}\, g)^\dagger \\
&= (Y \# [\eta^\nu, f^\ddagger h]^\bullet)\ \diamondsuit g. && /\!\!/ \text{ definition of } \diamondsuit
\end{aligned}$$

We thus obtained that $f^\ddagger h$ satisfies the fixpoint equation for $g^\ddagger$, hence $f^\ddagger h = g^\ddagger$. ◀

Recall that $T^\nu_H = F_\#$ for $X \# Y = T(X + HY)$.

▶ **Corollary 26.** *Given a guarded Elgot monad* $\mathbf{T}$*, then* $\mathbf{T}^\nu_H$ *is also guarded Elgot with the requisite structure obtained from the parametrized guarded Elgot monad* $X \# Y = T(X + HY)$*.*

5.2 Free Extensions of Guarded Elgot Monads

We proceed to apply the results of the previous section under $X \# Y = T(X + HY)$.

▶ **Lemma 27.** *Let* $\mathbf{T}$ *be a guarded monad.*

1. *The natural transformation*

$$HX \xrightarrow{\eta\, \mathsf{inr}(H\eta^\nu)} T(X + HT^\nu_H X) \xrightarrow{\mathsf{out}^{-1}} T^\nu_H X$$

is guarded.

2. *The natural transformation*

$$TX \xrightarrow{T\, \mathsf{inl}} T(X + HT^\nu_H X) \xrightarrow{\mathsf{out}^{-1}} T^\nu_H X$$

is a guarded Elgot monad morphism.

Proof. The first clause is obvious by definition. For the second clause, we need to check that the relevant morphism preserves guarded iteration, that is, given $f\colon X \to Y \rangle X$, we need to show that

$$\mathsf{out}^{-1}(T\,\mathsf{inl})\,f^\dagger = (\mathsf{out}^{-1}(T\,\mathsf{inl})\,f)^\ddagger.$$

By definition of $(-)^\ddagger$, equivalently, we prove the equation

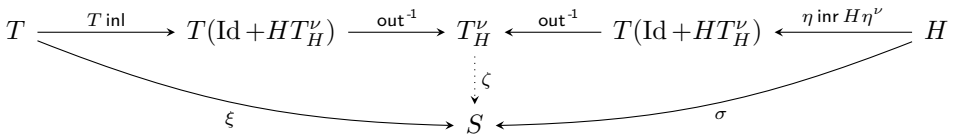
$$(T\,\mathsf{inl})\,f^\dagger = T(\mathsf{id} + [\eta^\nu, (T\,\mathsf{inl})\,f^\dagger]^\star)\diamondsuit(\mathsf{out}^{-1}(T\,\mathsf{inl})\,f).$$

Indeed,

$$\begin{aligned} &T(\mathsf{id} + [\eta^\nu, (T\,\mathsf{inl})\,f^\dagger]^\star)\diamondsuit(\mathsf{out}^{-1}(T\,\mathsf{inl})\,f) \\ &\qquad = T(\mathsf{id} + [\eta^\nu, (T\,\mathsf{inl})\,f^\dagger]^\star)\,(T(\mathsf{inl} + \mathsf{id})\,f)^\dagger && /\!/ \text{ definition of } \diamondsuit \\ &\qquad = T(\mathsf{id} + [\eta^\nu, (T\,\mathsf{inl})\,f^\dagger]^\star)\,(T\,\mathsf{inl})\,f^\dagger && /\!/ \text{ naturality} \\ &\qquad = (T\,\mathsf{inl})\,f, \end{aligned}$$

as desired. ◀

▶ **Theorem 28.** *In the category of guarded Elgot monads, $\mathbf{T}^\nu_H$, provided it exists, is a free extension of $\mathbf{T}$ by H in the sense of Definition 2, that is, for every guarded Elgot monad $\mathbf{S}$, a guarded Elgot monad morphism $\xi : \mathbf{T} \to \mathbf{S}$ and a guarded natural transformation $\sigma : H \to \mathbf{S}$, there exists a guarded Elgot monad morphism $\zeta : \mathbf{T}^\nu_H \to \mathbf{S}$ uniquely characterized by the following commutative diagram*



in the obvious category of natural transformations. Concretely, every $\zeta_X : T^\nu_H X \to SX$ has the form $\big(T^\nu_H X \xrightarrow{\xi\,\mathsf{out}} S(X + HT^\nu_H X) \xrightarrow{[\eta\,\mathsf{inl},\,(S\,\mathsf{inr})\,\sigma]^\star} S(X + T^\nu_H X)\big)^\dagger$.

Proof. By Lemma 27 the candidate monad morphism $\mathbf{T} \to \mathbf{T}^\nu_H$ and the candidate natural transformation $H \to \mathbf{T}^\nu_H$ are guarded. The trickiest part of the claim is the fact that $\mathbf{T}^\nu_H$ is indeed a guarded Elgot monad, which is shown in Theorem 25. Let us verify that ζ is a guarded monad morphism. Suppose that $f : X \to T^\nu_H(Y + X)$ is inr-guarded and show that $\zeta\,f : X \to S(Y + X)$ is inr-guarded. We have

$$\begin{aligned} \zeta\,f &= [\eta, \zeta]^\star[\eta\,\mathsf{inl},\ (S\,\mathsf{inr})\,\sigma]^\star\,\xi\,\mathsf{out}\,f && /\!/ \text{ fixpoint} \\ &= [\eta, \zeta^\star\sigma]^\star\,\xi\,\mathsf{out}\,f, \end{aligned}$$

which can be presented as

$$X \xrightarrow{\mathsf{out}\,f} T((Y + X) + HT^\nu_H(Y + X)) \cong T((Y + HT^\nu_H(Y + X)) + X) \xrightarrow{[[\eta\,\mathsf{inl},\zeta^\star\sigma],\eta\,\mathsf{inr}]^\star} S(Y + X).$$

The composite morphism in the upper row is inr-guarded by definition. We will be done by **(cmp)** if we show that the morphism $[\eta\,\mathsf{inl}, \zeta^\star\sigma]$ occurring in the lower row is inr-guarded. By **(trv)** and **(par)** this amounts to showing that $\zeta^\star\sigma : HT^\nu_H(Y + X) \to S(Y + X)$ is

inr-guarded. Indeed, σ is id-guarded by definition, hence $\zeta^\star\sigma$ is id-guarded by **(cmp)**, which weakens to inr-guardedness by Proposition 8.

The remaining calculations are the same as in previous work [9, Theorem 8.3] where the analogous statement was shown in the unguarded case. ◀

Note that the initial guarded Elgot monad is the identity monad under vacuous guardedness. Using Proposition 5 we obtain

▶ **Corollary 29.** *Given a guarded Elgot monad* T *and an endofunctor* H, T^ν_H *is the coproduct, in the category of guarded Elgot monads, of* T *and* $\nu\gamma.(- + H\gamma)$*, provided the latter exists.*

6 Conclusions and Further Work

We introduced guarded Elgot monads as a common generalization of Elgot monads and guarded iterative monads previously studied in the literature. We propose to use them as a yardstick for analyzing sophisticated notions of iteration when the iteration operator is neither total nor a unique solution of the corresponding fixpoint equation. Situations of this kind indeed occur in practice, e.g. in process semantics wrt infinite trace equivalence (Example 20 (5)). Moreover, guarded Elgotness tends to propagate along monad transformers, which leads to further examples. We explored one such monad transformer, receiving as an input a monad T and a functor H and returning a monad T^ν_H of possibly non-terminating processes under strong bisimilarity, with side-effects by T and with actions by H. Our main theorem shows that for a guarded Elgot monad T, T^ν_H is canonically guarded Elgot, and more specifically it can be characterized as a free extension of T by H in the category of all guarded Elgot monads.

The monad transformer $\mathsf{T} \mapsto \mathsf{T}^\nu_H$ is particularly important because the semantic domain it generates is subject to (coalgebraic) strong bisimilarity, which is arguably the finest semantic equivalence on processes. We thus hope to obtain further interesting generic process equivalences, most importantly infinite trace equivalence, in a principled fashion, as quotients of T^ν_H under suitably defined iteration-congruences. We plan to explore connections between the outlined approach to characterizing process equivalences and universal characterizations of such equivalences, such as the final coalgebra characterization of finite trace equivalence given in [4].

References

1. J. Adámek, R. Börger, S. Milius, and J. Velebil. Iterative algebras: How iterative are they? *Theory Appl. Cat.*, 19:61–92, 2008.
2. Stephen Bloom and Zoltán Ésik. *Iteration theories: the equational logic of iterative processes.* Springer, 1993.
3. F. Borceux. *Handbook of Categorical Algebra 1.* Cambridge University Press, 1994.
4. N. Bowler, P.B. Levy, and G.D. Plotkin. Initial algebras and final coalgebras consisting of nondeterministic finite trace strategies. In *Proceedings, 34th Conference on the Mathematical Foundations of Programming Semantics*, volume 341 of *ENTCS*, pages 23–44, 2018.
5. Venanzio Capretta. General recursion via coinductive types. *Log. Meth. Comput. Sci.*, 1(2), 2005.
6. Pietro Cenciarelli and Eugenio Moggi. A syntactic approach to modularity in denotational semantics. In *Category Theory and Computer Science, CTCS 1993*, 1993.
7. Sergey Goncharov, Stefan Milius, and Christoph Rauch. Complete Elgot Monads and Coalgebraic Resumptions. In *Mathematical Foundations of Programming Semantics, MFPS 2016*, volume 325 of *ENTCS*, pages 147–168. Elsevier, 2016.

8 Sergey Goncharov and Lutz Schröder. Guarded Traced Categories. In Christel Baier and Ugo Dal Lago, editors, *Proc. 21th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2018)*, volume 10803 of *LNCS*, pages 313–330. Springer, 2018.

9 Sergey Goncharov, Lutz Schröder, Christoph Rauch, and Julian Jakob. Unguarded Recursion on Coinductive Resumptions. *Logical Methods in Computer Science*, 14(3), 2018.

10 Sergey Goncharov, Lutz Schröder, Christoph Rauch, and Maciej Piróg. Unifying Guarded and Unguarded Iteration. In Javier Esparza and Andrzej Murawski, editors, *Foundations of Software Science and Computation Structures, FoSSaCS 2017*, volume 10203 of *LNCS*, pages 517–533. Springer, 2017.

11 Masahito Hasegawa. *Models of Sharing Graphs: A Categorical Semantics of Let and Letrec.* Springer, 1999.

12 Masihito Hasegawa. Recursion from Cyclic Sharing: Traced Monoidal Categories and Models of Cyclic Lambda Calculi. In *Typed Lambda Calculi and Applications, TLCA 1997*, volume 1210 of *LNCS*, pages 196–213. Springer, 1997.

13 Martin Hyland, Paul Levy, Gordon Plotkin, and John Power. Combining algebraic effects with continuations. *Theoret. Comput. Sci.*, 375(1-3):20–40, 2007.

14 Martin Hyland, Gordon Plotkin, and John Power. Combining Computational Effects: Commutativity & Sum. In *TCS'02*, volume 223, pages 474–484. Kluwer, 2002.

15 Martin Hyland, Gordon Plotkin, and John Power. Combining effects: Sum and tensor. *Theoret. Comput. Sci.*, 357(1-3):70–99, 2006.

16 Stefan Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput.*, 196(1):1–41, 2005.

17 R. Milner. *Communication and concurrency.* Prentice-Hall, 1989.

18 Eugenio Moggi. Notions of Computation and Monads. *Inf. Comput.*, 93:55–92, 1991.

19 Maciej Piróg and Jeremy Gibbons. The Coinductive Resumption Monad. In *Mathematical Foundations of Programming Semantics, MFPS 2014*, volume 308 of *ENTCS*, pages 273–288, 2014.

20 Maciej Piróg and Jeremy Gibbons. Monads for Behaviour. In *Mathematical Foundations of Programming Semantics, MFPS 2013*, volume 298 of *ENTCS*, pages 309–324, 2015.

21 A. W. Roscoe. Unbounded nondeterminism in CSP. Technical Report PRG-67, Oxford University Computing Laboratory, July 1988. in *Two papers on CSP*, Also appeared in Journal of Logic and Computation, Vol 3, No 2 pp131-172 (1993). URL: `http://www.cs.ox.ac.uk/people/bill.roscoe/publications/28.ps`.

22 Alex Simpson and Gordon Plotkin. Complete axioms for categorical fixed-point operators. In *Logic in Computer Science, LICS 2000*, pages 30–41, 2000.

23 Tarmo Uustalu. Generalizing Substitution. *ITA*, 37(4):315–336, 2003.